Noname manuscript No.
(will be inserted by the editor)

Online classifier adaptation for cost-sensitive learning

Junlin Zhang · José García

the date of receipt and acceptance should be inserted later

Abstract In this paper, we propose the problem of online cost-sensitive classifier adaptation and the first algorithm to solve it. We assume we have a base classifier for a cost-sensitive classification problem, but it is trained with respect to a cost setting different to the desired one. Moreover, we also have some training data samples streaming to the algorithm one by one. The problem is to adapt the given base classifier to the desired cost setting using the steaming training samples online. To solve this problem, we propose to learn a new classifier by adding an adaptation function to the base classifier, and update the adaptation function parameter according to the streaming data samples. Given a input data sample and the cost of misclassifying it, we update the adaptation function parameter by minimizing cost weighted hinge loss and respecting previous learned parameter simultaneously. The proposed algorithm is compared to both online and off-line cost-sensitive algorithms on two cost-sensitive classification problems, and the experiments show that it not only outperforms them one classification performances, but also requires significantly less running time.

Keywords Cost-sensitive learning · Classifier adaptation · Online learning · Fact detection · Car detection

J. Zhang*
Xi'an University of Architecture and Technology, Xi'an, Shaanxi 710055, P.R. China
E-mail: junlinzhang1@yahoo.com
*J. Zhang is the corresponding author.

J. García
Department of Computer Languages and Systems, Universitat Jaume I, Av. Sos Baynat s/n, 12071 Castelío de la Plana, Spain
E-mail: josegarciajaumei@outlook.com

1 Introduction

In pattern recognition problems, we try to design a classification function to predict the class label of a data sample, so that the misclassification errors of a set of training samples can be minimized [3,11,12,20,30,4]. A popular assumption for the learning of a classifier is that the loss of misclassifying any data sample in the training set is equal. However, in real-world applications, different misclassifications may result in significant different costs. For example, in the problem of breast cancer diagnosis, misclassifying a malignant tumor sample may cause much more cost than misclassifying a benign tumor sample. Thus it is necessary to take the costs of different types of misclassifications into account when a classifier is trained. This problem is named cost-sensitive learning in machine learning community [33,16,48,13,58]. Given the cost setting, i.e., costs of different misclassifications, the target of cost-sensitive learning is to train a classifier so that the cost of overall misclassification can be minimized. In cost-sensitive binary classification, we can have different costs for misclassifications of positive and negative samples. In this case, misclassifying a positive sample to a negative sample incorrectly may results much higher cost than misclassifying a negative sample to a positive sample. So we must design a classify to correctly classify most of the positive samples, while allow some misclassification of negative samples. In this way, the overall misclassification cost can be minimized. Lots of cost-sensitive learning algorithms have been proposed to take account of different misclassification costs. For example, Zhou et al. [58] proposed to train cost-sensitive neural networks by using technologies of sampling and threshold-moving (STM), so that the distribution of the training data samples can be modify, and the costs of different types of misclassifications can be conveyed by the appearances of the examples. Sun et al. [32] provided a comprehensive analysis of the AdaBoost algorithm regarding its application in the class imbalance problem, and developed three cost-sensitive boosting algorithms (CSB), by introducing cost items into the learning framework of AdaBoost. Masnadi-Shirazi and Vasconcelos [25] also proposed a AdaBoost-based cost-sensitive learning algorithm (ABC) to design cost-sensitive boosting algorithms, by considering two necessary conditions for optimal cost-sensitive learning, which are the minimization of expected losses by optimal cost-sensitive decision rules, and the minimization of empirical loss to emphasize the neighborhood of the desired cost-sensitive boundary. Ting [34] introduced a sample-weighting method (SW) to induce cost-sensitive trees, by generalizing the standard tree induction process and initial instance weights determine the type of tree to be induced-minimum error trees or minimum high cost error trees. Chen eta al. [10] proposed a supervised learning algorithm fast flux discriminant, for large-scale nonlinear cost-sensitive classification problems, by decomposing the kernel density estimation in the original feature space into selected low-dimensional subspaces. This method archives the efficiency, interpretability and accuracy simultaneously, and meanwhile it is also sparse and naturally handles mixed data types.

With rapid development of internet technology, more and more data is generated continuously, and the training set of data samples is been increased every day with new data samples added to the training set. Moreover, the cost setting can also be changed from time to time. This proposed tow new challenges to the cost sensitive learning problems:

1. When the cost setting is changed, the learned classifier cannot be adapted to the new cost setting. A possible strategy to solve this problem is to learn a new classifier according to the new cost setting using the entire training set from the very beginning and ignores the previous learned classifier with previous cost setting. However this strategy is time-consuming, especially when the training set is large.
   When we already have a classifier learned according to a cost setting, can we utilize it to learn another classifier with regard to a different cost setting? This problem is defined as classifier adaption. Actually, classifier adaptation has been applied to performance measures optimization [21] and cross-domain learning [47]. In [21], a classifier is learned to optimize a performance measure, and then adapted to optimize another performance measure, while in [47], a classifier is learned from a domain, and then adapted to a different domain. In this paper, we propose the problem of adapting a learned classifier to a different cost setting.
2. When the data samples are generated and added to the training set one by one, the transitional cost sensitive learning methods cannot be applied, since they assumes that the entire training set is given to the algorithm once. Recently, cost-sensitive online classification (CSOC) method was proposed by Wang et al. [39]. This method takes the training set one by one and update the cost sensitive classifier online [29,49,14]. However, CSOC is also constrained to fixed cost setting. When a cost setting is given, it learn a new classifier online, and ignores the other classifiers learned with different cost settings. Can we learn a classifier from a base classifier trained with different cost sensitive setting online? This problem remains an open problem.

To solve the above two problems simultaneously, in this paper, we propose the first online cost-sensitive classier adaption method. We assume that we have a existed cost-sensitive classifier, and we try to adapt it to another classifier with regard to a different cost setting, with help of data samples appearing one by one in an online way. The adaptation is implemented by adding an adaptation function, and the it is learned by updating the adaptation function parameter with the coming training samples with different misclassification costs. We construct an objective function by considering the respecting previous learned and minimizing cost weighted hinge loss with coming training samples. By solving the objective function with a gradient descent method and we develop an iterative algorithm. The contributions of this paper are of two folds:

1. We proposed the problem of online cost-sensitive classifier adaptation.

2. We proposed a novel algorithm to solve this problem.

The rest parts of this paper are organized as follows: in Section 2, we introduce the proposed novel method. In Section 3, the proposed method is evaluated on some benchmark data sets. In Section 4, the paper is concluded with some future works.

2 Proposed Method

2.1 Problem Formulation

In this paper, instead of learning a novel cost-sensitive classifier from the given training set and the cost setting, we hope to use the existed classifier by employing the framework of classifier adaptation to learn the cost-sensitive classifier effectively. Suppose that we already have a classifier $f_0(\mathbf{x})$ learned without consider the different costs of misclassifications of positive and negative samples, or a classifier learned with different cost setting, we want to adapt it to a problem with a new cost setting. To this end, we construct a new classifier $f(\mathbf{x})$ by adding a linear adaptation function $\mathbf{w}^\top \mathbf{x}$ to $f_0(\mathbf{x})$, i.e.

$$f(\mathbf{x}) = f_0(\mathbf{x}) + \mathbf{w}^\top \mathbf{x} \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^d$ is the adaptation function parameter. Please note that $f_0(\mathbf{x})$ can be any type of classifier, for example, SVM, Adaboost, etc. In this way, we transfer the problem of cost-sensitive classifier adaptation to the learning of $\mathbf{w}$.

In the traditional cost-sensitive learning problem, a training data set composed of many positive and negative training samples are given. The cost factors of misclassification of positive and negative samples are denoted as C_+ and C_- respectively. Please note that when we train $f_0(\mathbf{x})$, C_+ and C_- are set to different values. The target of cost-sensitive learning is to learn a classifier which could minimize the overall cost of misclassification of the training samples. However, in the online learning scene, we do not have the entire training data set during the training procedure. Instead, the training data samples are given sequentially, and the algorithm is run in an iterative way. In each iteration, only one training sample is given, and the classifier is updated only with regard to this training sample. In the t-th iteration, we assume that wa have a training sample $(\mathbf{x}_t, y_t)$, where $\mathbf{x}_t \in \mathbb{R}^d$ is its d-dimensional feature vector, and $y_t \in \{+1, -1\}$ is its corresponding class label. The corresponding misclassification cost is also given as C_t,

$$C_t = \begin{cases} C_+, if\ y_i = +1; \\ C_-, if\ y_i = -1. \end{cases} \tag{2}$$

Moreover, we also assume that we already learned an adaptation function parameter from the previous iteration $\mathbf{w}_{t-1}$. To update $\mathbf{w}$, we consider the following two problems.

- **Respecting previous learned $\mathbf{w}_{t-1}$**: To make the learned $\mathbf{w}$ consistent, we hope the updated $\mathbf{w}_t$ to respect the previous $\mathbf{w}_{t-1}$. To this end, we minimize the squared ℓ_2 distance between them,

$$\min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w} - \mathbf{w}_{t-1}\|_2^2 . \tag{3}$$

- **Minimizing Cost Weighted Hinge Loss**: To measure the loss of misclassification, we apply the hinge loss function to $(\mathbf{x}_t, y_t)$, which is defined as

$$\begin{aligned} L(y_t, f(\mathbf{x}_t)) &= \max(0, 1 - y_t f(\mathbf{x}_t)) \\ &= \max\left(0, 1 - y_t \left(f_0(\mathbf{x}_t) + \mathbf{w}^\top \mathbf{x}_t\right)\right) . \end{aligned} \tag{4}$$

  Since positive and negative samples have different misclassification costs, we weight the hinge loss of the t-th sample by its corresponding cost factor C_t, and minimize the weighted loss,

$$\min_{\mathbf{w}} C_t \times \max(0, 1 - y_t \left(f_0(\mathbf{x}_t) + \mathbf{w}^\top \mathbf{x}_t\right)) \tag{5}$$

  By introducing a nonnegative slack variable ξ, the optimization problem with a cost weighted hinge loss is transferred to

$$\begin{aligned} &\min_{\mathbf{w},\xi} C_t \xi, \\ &s.t.\ 1 - y_t \left(f_0(\mathbf{x}_t) + \mathbf{w}^\top \mathbf{x}_t\right) \leq \xi, 0 \leq \xi. \end{aligned} \tag{6}$$

  Here the cost factor C_t is similar to the penalty factor of SVM. However, we must note that this penalty factor is cost-sensitive.

By considering the problems in (3) and (6) simultaneously, we obtain the optimization problem for the updating of $\mathbf{w}$ in the t-th iteration,

$$\begin{aligned} (\mathbf{w}_t, \xi_t) = \arg\min_{\mathbf{w},\xi} &\frac{1}{2} \|\mathbf{w} - \mathbf{w}_{t-1}\|_2^2 + \alpha C_t \xi, \\ &s.t.\ 1 - y_t \left(f_0(\mathbf{x}_t) + \mathbf{w}^\top \mathbf{x}_t\right) \leq \xi, 0 \leq \xi. \end{aligned} \tag{7}$$

where α is a tradeoff parameter, and it is chosen by cross-fold validation on a training set. By solving this problem, we can obtain an adaptation function parameter $\mathbf{w}_t$ with regard to the training sample input in the t-th iteration. We should note that the obtained $\mathbf{w}_t$ is learned from both the previous $\mathbf{w}_{t-1}$ and the training sample $(\mathbf{x}_t, y_t)$. Most importantly, the updating of $\mathbf{w}_t$ relies on the cost of misclassification cost of $(\mathbf{x}_t, y_t)$ by considering the cost factor as a loss weight. When different samples come, different cost factor is used and the hinge loss is weighted correspondingly.

2.2 Optimization

To optimize the objective function in (7), we use the Lagrange multiplier method. The Lagrange function is

$$\begin{aligned}\mathcal{L}(\mathbf{w},\xi,\tau,\lambda) =& \frac{1}{2}\left\|\mathbf{w}-\mathbf{w}_{t-1}\right\|_2^2+\alpha C_t\xi \\ &+\tau\left(1-y_t\left(f_0(\mathbf{x}_t)+\mathbf{w}^\top\mathbf{x}_t\right)-\xi\right)-\lambda\xi.\end{aligned} \tag{8}$$

where τ is the nonnegative Lagrange multiplier for the constrain of $1-y_t\left(f_0(\mathbf{x}_t)+\mathbf{w}^\top\mathbf{x}_t\right)\leq\xi$, and λ is the nonnegative Lagrange multiplier for the constrain of $0\leq\xi$. According to the dual theory of optimization, the minimization of (7) can be achieved by solving the following dual problem,

$$\begin{aligned}&\max_{\tau,\lambda}\min_{\mathbf{w},\xi}\ \mathcal{L}(\mathbf{w},\xi,\tau,\lambda)\\ &\qquad s.t.\ \tau\geq 0,\lambda\geq 0.\end{aligned} \tag{9}$$

To solve this problem, we set the divertive of the Lagrange function $\mathcal{L}(\mathbf{w},\xi,\tau,\lambda)$ with regard to $\mathbf{w}$ to zero, and we have

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}}=(\mathbf{w}-\mathbf{w}_{t-1})-\tau y_t\mathbf{x}_t=0\Rightarrow\mathbf{w}=\mathbf{w}_{t-1}+\tau y_t\mathbf{x}_t. \tag{10}$$

Remark: The motivation of setting the divertive of the Lagrange function with regard to $\mathbf{w}$ to zero is to solve the inner minimization problem. In equation (9), the problem is coupled with two optimization problems, which is a inner minimization problem, and an outer maximization problem. To solve this coupled problem, our strategy is first solving the inner problem, and then substituting the results to the objective to solve the outer problem. According to the optimization theory, the minimization of $\mathcal{L}$ with regard to $\mathbf{w}$ is reached at a solution making its divertive zero, thus we should set the divertive of the $\mathcal{L}$ with regard to $\mathbf{w}$ to zero to obtain the optimal $\mathbf{w}$.

Moreover, we also set its divertive with regard to ξ to zero, and obtain

$$\frac{\partial\mathcal{L}}{\partial\xi}=\alpha C_t-\tau-\lambda=0\Rightarrow\alpha C_t-\tau=\lambda\geq 0\Rightarrow\tau\leq\alpha C_t. \tag{11}$$

Substituting results of both (10) and (11) to the Lagrange function in (8), we can rewrite it as the function of only variable τ,

$$\begin{aligned}\mathcal{L}(\tau)=&\frac{1}{2}\left\|\tau y_t\mathbf{x}_t\right\|_2^2+\tau\left[1-y_t\left(f_0(\mathbf{x}_t)+\left(\mathbf{w}_{t-1}+\tau y_t\mathbf{x}_t\right)^\top\mathbf{x}_t\right)\right]\\ =&\frac{1}{2}\tau^2\mathbf{x}_t^\top\mathbf{x}_t+\tau\left[1-y_t\left(f_0(\mathbf{x}_t)+\mathbf{w}_{t-1}^\top\mathbf{x}_t\right)\right]-\tau^2\mathbf{x}_t^\top\mathbf{x}_t\\ =&-\frac{1}{2}\tau^2\mathbf{x}_t^\top\mathbf{x}_t+\tau\left[1-y_t\left(f_0(\mathbf{x}_t)+\mathbf{w}_{t-1}^\top\mathbf{x}_t\right)\right].\end{aligned} \tag{12}$$

By setting the divertive of $\mathcal{L}(\tau)$ with regard to τ to zero, we have the initial solution of τ,

$$
\begin{aligned}
&\frac{\partial \mathcal{L}(\tau)}{\partial \tau} = -\tau \mathbf{x}_t^\top \mathbf{x}_t + \left[1 - y_t\left(f_0(\mathbf{x}_t) + \mathbf{w}_{t-1}^\top \mathbf{x}_t\right)\right] = 0\\
&\Rightarrow \tau = \frac{1 - y_t\left(f_0(\mathbf{x}_t) + \mathbf{w}_{t-1}^\top \mathbf{x}_t\right)}{\mathbf{x}_t^\top \mathbf{x}_t}.
\end{aligned}
\tag{13}
$$

Moreover, we should also note that in (8), we have a constrain $\tau \geq 0$, and in (11) we have another constrain $\tau \leq \alpha C_t$. Thus the solution of τ must fall in the following range,

$$
0 \leq \tau \leq \alpha C_t. \tag{14}
$$

In this way, the solution of τ^t can be obtained by discussing the following three cases:

1. **Case I**: When $\frac{1-y_t\left(f_0(\mathbf{x}_t)+\mathbf{w}_{t-1}^\top \mathbf{x}_t\right)}{\mathbf{x}_t^\top \mathbf{x}_t} \leq 0$, the solution of τ_t is

   $$
   \tau_t = 0, \tag{15}
   $$

   so that the constrain $\tau \geq 0$ can be satisfied.
2. **Case II**: When $0 < \frac{1-y_t\left(f_0(\mathbf{x}_t)+\mathbf{w}_{t-1}^\top \mathbf{x}_t\right)}{\mathbf{x}_t^\top \mathbf{x}_t} \leq \alpha C_t$, the solution of τ_t is

   $$
   \tau_t = \frac{1 - y_t\left(f_0(\mathbf{x}_t) + \mathbf{w}_{t-1}^\top \mathbf{x}_t\right)}{\mathbf{x}_t^\top \mathbf{x}_t}, \tag{16}
   $$

   so that the minimization of (7) can be archived.
3. **Case III**: When $\alpha C_t < \frac{1-y_t\left(f_0(\mathbf{x}_t)+\mathbf{w}_{t-1}^\top \mathbf{x}_t\right)}{\mathbf{x}_t^\top \mathbf{x}_t}$, we have the solution of τ_t as

   $$
   \tau_t = \alpha C_t, \tag{17}
   $$

   so that the constrain $\tau_t \leq \alpha C_t$ can be satisfied.

After τ_t is determined, we can then update $\mathbf{w}_t$ using the result in (10) as follows,

$$
\mathbf{w}_t = \mathbf{w}_{t-1} + \tau_t y_t \mathbf{x}_t. \tag{18}
$$

It could be note that the new classifier adaptation function parameter is obtained by adding a bias term $y_t \mathbf{x}_t$ determined by the t-th sample to the previous $\mathbf{w}_{t-1}$. The bias term is weighted by a Lagrange multiplier τ_t which is further controlled by the cost factor of the t-th sample.

2.3 Algorithm

Based on the optimization results, we can develop an online cost-sensitive classifier adaptation algorithm which can take training samples one by one. The algorithm takes an initial classifier $f(\mathbf{x})$ as an input, and operates on a iterative way. In each iteration, one new training sample is input to update the classifier adaptation function parameter, based on the updating rules in (15) - (17), and (18).

Algorithm 1 Online Cost-Sensitive Classifier Adaptation algorithm (OC-SCA).

```
Input: An initial classifier function $f_0(\mathbf{x})$;
Input: Tradeoff parameter $\alpha$;
Initialize $t = 0$ and $\mathbf{w}_0 = \mathbf{0}$
while A new training sample $(\mathbf{x}_t, y_t)$ with its corresponding misclassification cost $C_t$ is input do
    Compute the initial solution of Lagrange multiplier $\tau$ as
        $$\tau_t' = \frac{1 - y_t\left(f_0(\mathbf{x}_t) + \mathbf{w}_{t-1}^\top \mathbf{x}_t\right)}{\mathbf{x}_t^\top \mathbf{x}_t}. \quad (19)$$
    Update $\tau_t$ as
        $$\tau_t = \begin{cases} 0, if\ \tau_t' \leq 0, \\ \tau_t', if\ 0 < \tau_t' \leq \alpha C_t \\ \alpha C_t, if\ \tau_t' > \alpha C_t. \end{cases} \quad (20)$$
    Update $\mathbf{w}_t$ as
        $$\mathbf{w}_t = \mathbf{w}_{t-1} + \tau_t y_t \mathbf{x}_t. \quad (21)$$
    Update $t = t + 1$;
end while
Output: Output the learned cost-sensitive classifier function $f(\mathbf{x}) = f_0(\mathbf{x}) + \mathbf{w}_{t-1}{}^\top \mathbf{x}$
```

3 Experiments

In this section, we studied the proposed algorithm experimentally.

3.1 Data sets

In the experiments, we used two cost-sensitive learning data sets, which are introduced as follows.

3.1.1 Face detection data set

The first data set is a face detection data set used in [35]. This data set is a large data set, and it contains 9832 face images and 9832 non-face images. Each face image is treated as a positive sample, while each non-face image is treated as a negative sample. Moreover, each image is represented as 50,000 dimensional visual feature vector. The problem of face detection is to classify a given candidate image to face or non-face. Moreover, we set the cost of misclassifying a face to non-face as 5, and that of misclassifying a non-face to fact to 1.

3.1.2 Car detection data set

The second data set we used is a car detection data set [1]. This data set contains 500 car images and 500 non-car images. The problem of car detection is to classify a given candidate image to car or non-car. In this problem car images are defined as positive images, and the non-car images are defined as negative images. In this case, we set the cost of misclassification of a car image to 8, and that of a non-car image to 1.

3.2 Experiment setup

To conduct the experiment, we used the 10-fold cross validation. An entire data set was split into 10 folds randomly, and then each set is used as a test set, while the remaining 9 sets were combined as a training set. Moreover, since the proposed method is based on the adaption of a classifier $f_0(\mathbf{x})$ trained with different cost setting, we further split the training set to two subsets. The first subset contains 2 folds, and we used it to train $f_0(\mathbf{x})$ with different cost setting. For the first data set, we used the cost setting of $C_+ = 2$ and $C_- = 1$ to train $f_0(\mathbf{x})$, and for the second data set, used $C_+ = 3$ and $C_- = 1$. The second subset contains 7 folds, and we used it to learn $\mathbf{w}$ using the proposed online learning algorithm, by inputting the training samples of the second training subset to the algorithm one by one.

The classification performances were measured by the average classification accuracies and the average misclassification costs. They are defined as follows,

$$\begin{aligned}
Average\ classification\ accuracy &= \frac{\sum_{i:\mathbf{x}_in\in\mathcal{T}} I(y_i = y_i^*)}{\sum_{i:\mathbf{x}_i\in\mathcal{T}} 1},\\
Average\ misclassification\ cost &= \frac{\sum_{i:\mathbf{x}_in\in\mathcal{T}} C_i I(y_i \neq y_i^*)}{\sum_{i:\mathbf{x}_i\in\mathcal{T}} 1},
\end{aligned} \tag{22}$$

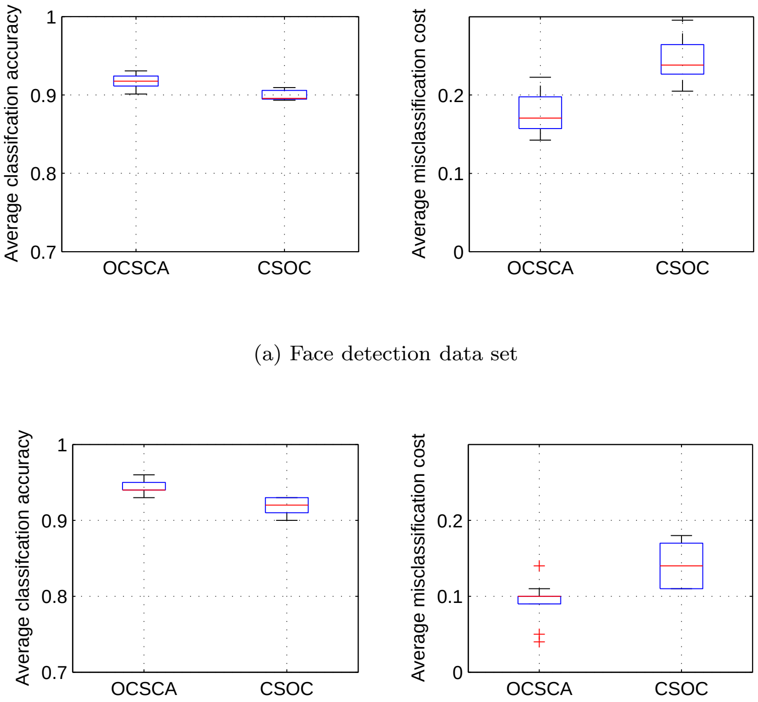
where $\mathcal{T}$ is the test set, y_i^* is the predicted class label, and $I(y_i = y_i^*) = 1$ if $y_i = y_i^*$, and 0 otherwise.

3.3 Results

We first compared the purposed online cost sensitive learning algorithm based on classifier adaptation to an online cost sensitive learning algorithm without considering the existed classifier $f_0(\mathbf{x})$, and then compared it to some transitional cost sensitive learning algorithm. Because the proposed algorithm is the only method that can take advantage of $f_0(\mathbf{x})$, for fear comparison, when we used the other algorithm, both $f_0(\mathbf{x})$ and the 2 folds in the training set used to train $f_0(\mathbf{x})$ were ignored.

3.3.1 Comparison to online cost sensitivity classification method

The boxplots of the classification accuracies and misclassification of the proposed online cost-sensitive classifier adaptation algorithm (OCSCA) and the CSOC algorithm over 10-fold cross validation are given in Fig. 1. From this figure, we can see that the proposed method outperforms the CSOC algorithm on both average accuracy and misclassification cost. Especially in the case of misclassification cost, the proposed algorithm achieves completely lower average misclassification cost then CSOC. This is because the proposed method takes advantage of an existing predictor learned from more data points by adapting it to a given cost setting. Even the existing predictor is learned according to a different cost setting. This is an strong evidence of the fact that classifier adaptation can benefit cost sensitive learning.



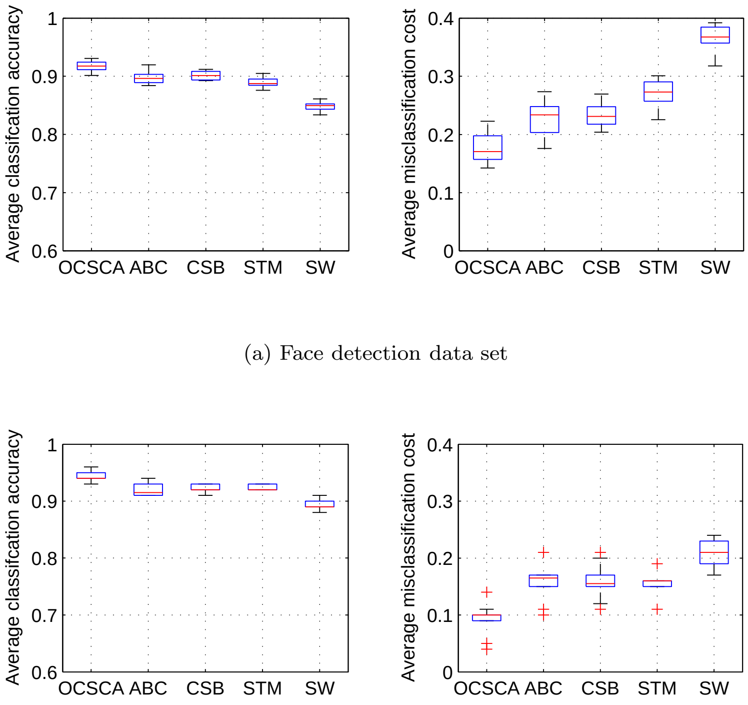
(a) Face detection data set

(b) Car detection data set

Fig. 1 Classification accuracies and misclassification costs of two online cost learning algorithms.

3.3.2 Comparison to off-line cost sensitivity classification method

We also compare the proposed OCSCA to four most popular off-line cost-sensitive learning algorithms, which are STM proposed by Zhou et al. [58], CSB proposed by Sun et al. [32], ABC proposed by Masnadi-Shirazi and Vasconcelos [25], and SW proposed by Ting [34]. The boxplots the classification accuracies and misclassification costs are given in Fig. 2. It is clear that in both the two figures, the proposed algorithm outperforms the compared algorithms on both classification accuracies and misclassification costs. The outperforming is even more significant on the misclassification costs. A main reason for this phenomenon lies on the fact that the proposed OCSCA algorithm starts learning from a base classifier $f_0(\mathbf{x})$, and then adapt it to the given cost setting via a training set, while the rest algorithms ignores $f_0(\mathbf{x})$ and directly learn the classifier from the training set. This means using a base classifier and adapting it to a training set can significantly boost the performance of cost-sensitive learning. Moreover, among the compared algorithms, it seems ABC and CSB performs slightly better than the other two ones. A possible reason is that they use the formula of Adaboost algorithm [31, 26, 2, 9], which performs well on detection problems.



(a) Face detection data set

(b) Car detection data set

Fig. 2 Classification accuracies and misclassification costs of the proposed algorithm and off-line cost-sensitive learning algorithms.

3.3.3 Running time

An important advantage of the proposed OCSCA algorithm is its low time complexity compared to off-line algorithms. Thus we also compared the running time of these methods and the results are given in Fig. 3. It is obverse that the running time of the two online learning algorithms OCSCA and CSOC is much less than that of the off-line learning algorithms. Both of OCSCA and CSOC take less than 200 seconds, while all the off-line learning algorithms take more than 800 seconds. This is not surprising because in each iteration, OCSCA and CSOC update the classifier using only one data sample, while the off-line learning algorithms needs to consider all the training samples.

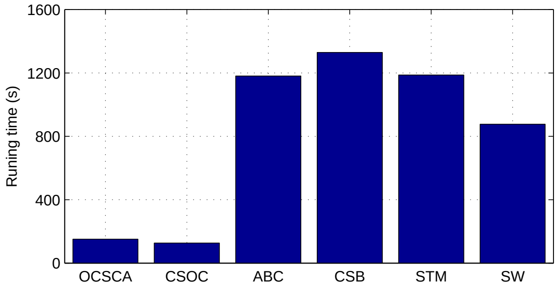


Fig. 3 Running time of learning procedure of online learning algorithms and off-line algorithms.

4 Conclusions and future works

In this paper, we propose the problem of adapting an existing base classifier to a cost-sensitive classification problem. The base classifier is trained using different cost settings. Moreover, we proposed a novel online learning algorithm for the adaptation of the classifier. The algorithm takes one data sample at one time to update the adaptation parameter. The advantages of this method are of two folds:

1. It can use the base classifier to boost the classification performance, and
2. its running time is low due to its online learning nature.

In this work, we used the SVM as the formulation of learning. In the future, we will study other classification methods, such as Adaboost. We will design an iterative algorithm to learn the classifier online by adapting an existing classifier trained with a different cost setting, and the adaptation function is an combination of some candidate weak classifiers. In each iteration, we have select a weak classifier according to the classification cost of the coming data point, and update its weight. Moreover, the loss function of Adaboost will be modified to consider the classification costs. Moreover, we will also investigate the application of the proposed algorithm to information security [41,43,53,42,41,52,18], bioinformatics [40,24,57], medial imaging [56,55,54], computer vision [38,23,22,36,37,19,17,46,44,45,15], reinforcement learning [27,28], cloud computing [50,51] and microprocessor reliability modeling [8,6,7,5,59,61,60].

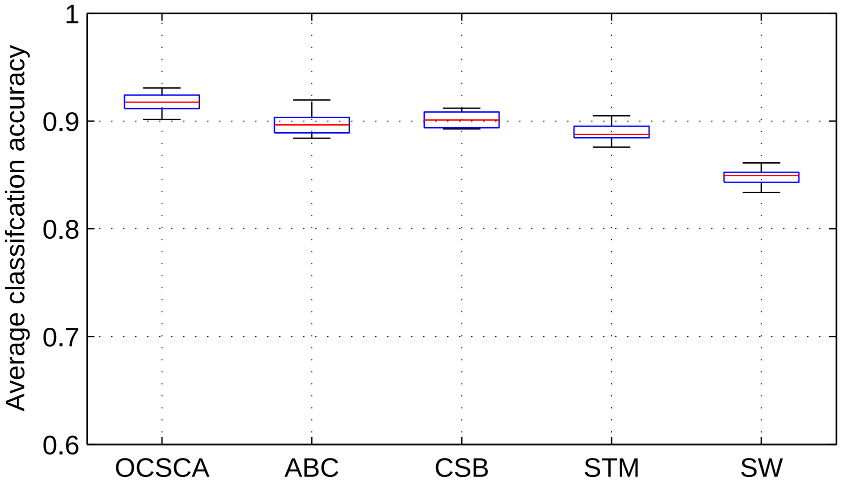
References

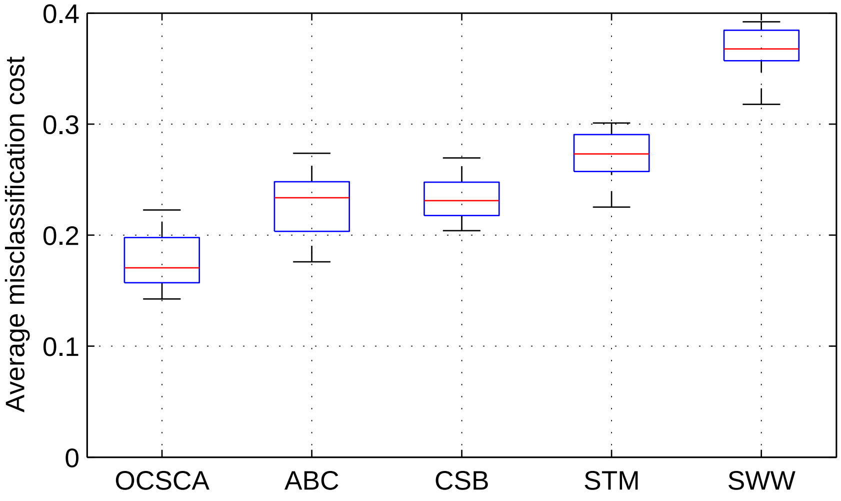
1. Agarwal, S., Awan, A., Roth, D.: Learning to detect objects in images via a sparse, part-based representation. Pattern Analysis and Machine Intelligence, IEEE Transactions on **26**(11), 1475–1490 (2004)
2. Ahachad, A., Omari, A., Figueiras-Vidal, A.: Neighborhood guided smoothed emphasis for real adaboost ensembles. Neural Processing Letters (2014)
3. Bischl, B., Mersmann, O., Trautmann, H., Preuss, M.: Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. In: GECCO'12 - Proceedings of the 14th International Conference on Genetic and Evolutionary Computation, pp. 313–320 (2012)
4. Charnay, C., Lachiche, N., Braud, A.: Pairwise optimization of bayesian classifiers for multi-class cost-sensitive learning. In: Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI, pp. 499–505 (2013)
5. Chen, C.C., Ahmed, F., Kim, D.H., Lim, S.K., Milor, L.: Backend dielectric reliability simulator for microprocessor system. Microelectronics Reliability **52**(9), 1953–1959 (2012)
6. Chen, C.C., Ahmed, F., Milor, L.: Impact of nbti/pbtion srams within microprocessor systems: Modeling, simulation, and analysis. Microelectronics Reliability **53**(9), 1183–1188 (2013)
7. Chen, C.C., Bashir, M., Milor, L., Kim, D.H., Lim, S.K.: Simulation of system backend dielectric reliability. Microelectronics Journal **45**(10), 1327–1334 (2014)
8. Chen, C.C., Milor, L.: Microprocessor aging analysis and reliability modeling due to back-end wearout mechanisms. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on **PP**(99), 1–1 (2014)
9. Chen, S.H., Pan, J.S., Lu, K., Xu, H.: Driving behavior analysis of multiple information fusion based on adaboost. Advances in Intelligent Systems and Computing **329**, 277–285 (2015)
10. Chen, W., Chen, Y., Weinberger, K.Q.: Fast flux discriminant for large-scale sparse nonlinear classification. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 621–630. ACM (2014)
11. Cherkassky, V., Ma, Y.: Practical selection of svm parameters and noise estimation for svm regression. Neural networks **17**(1), 113–126 (2004)
12. Dhanalakshmi, P., Palanivel, S., Ramalingam, V.: Pattern classification models for classifying and indexing audio signals. Engineering Applications of Artificial Intelligence **24**(2), 350–357 (2011)
13. Elkan, C.: The foundations of cost-sensitive learning. In: IJCAI International Joint Conference on Artificial Intelligence, pp. 973–978 (2001)
14. Fonseca, G., Destro-Filho, J., Vieira Borges, T.: Online learning system for biomedical engineering. In: 2007 IWSSIP and EC-SIPMCS - Proc. 2007 14th Int. Workshop on Systems, Signals and Image Processing, and 6th EURASIP Conf. Focused on Speech and Image Processing, Multimedia Communications and Services, pp. 169–172 (2007)
15. G.Guo, X.Wang: Kinship measurement on salient facial features. IEEE Transactions on Instrumentation and Measurement **61**(8), 2322–2325 (2012)
16. Greiner, R., Grove, A., Roth, D.: Learning cost-sensitive active classifiers. Artificial Intelligence **139**(2), 137–174 (2002)
17. Hao, J., Cai, W.: Trusted Block as a Service: Towards Sensitive Applications on the Cloud. In: Proceedings of the 10th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, pp. 73–82 (2011)

18. Hao, J., Cai, W.: Measuring Information Exposure Attacks on Interest Management. In: Proceedings of the 2012 ACM/IEEE/SCS 26th Workshop on Principles of Advanced and Distributed Simulation, PADS '12, pp. 133–144 (2012)
19. Hao, J., Liu, Y., Cai, W., Bai, G., Sun, J.: vTRUST: A Formal Modeling and Verification Framework for Virtualization Systems. In: Formal Methods and Software Engineering, vol. 8144, pp. 329–346 (2013)
20. Kusner, M., Chen, W., Zhou, Q., Xu, Z., Weinberger, K., Chen, Y.: Feature-cost sensitive learning with submodular trees of classifiers. In: Proceedings of the National Conference on Artificial Intelligence, vol. 3, pp. 1939–1945 (2014)
21. Li, N., Tsang, I., Zhou, Z.H.: Efficient optimization of performance measures by classifier adaptation. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(6), 1370–1382 (2013)
22. Li, X., Xu, H., Wan, S., Yin, Z., Yu, W.: Feature-aligned harmonic volumetric mapping using mfs. Computers & Graphics **34**(3), 242–251 (2010)
23. Li, X., Yin, Z., Wei, L., Wan, S., Yu, W., Li, M.: Symmetry and template guided completion of damaged skulls. Computers & Graphics **35**(4), 885–893 (2011)
24. Liu, Y., Yang, J., Zhou, Y., Hu, J.: Structure design of vascular stents. Multiscale Simulations and Mechanics of Biological Materials pp. 301–317 (2013)
25. Masnadi-Shirazi, H., Vasconcelos, N.: Cost-sensitive boosting. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33**(2), 294–309 (2011)
26. Mei, S., Zhu, H.: Adaboost based multi-instance transfer learning for predicting proteome-wide interactions between salmonella and human proteins. PLoS ONE **9**(10) (2014)
27. Min, H., Zeng, J., Chen, J., Zhu, J.: A study of reinforcement learning in a new multiagent domain. In: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 02, WI-IAT '08, pp. 154–161. IEEE Computer Society, Washington, DC, USA (2008)
28. Min, H., Zeng, J., Luo, R.: Fuzzy cmac with automatic state partition for reinforcementlearning. In: Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation, GEC '09, pp. 421–428. ACM, New York, NY, USA (2009)
29. Patil, P., Fatangare, Y., Kulkarni, P.: Semi-supervised learning algorithm for online electricity data streams. Advances in Intelligent Systems and Computing **324**, 349–358 (2015)
30. Persello, C., Dalponte, M., Gobakken, T., Naesset, E.: Optimizing the ground sample collection with cost-sensitive active learning for tree species classification using hyperspectral images. In: International Geoscience and Remote Sensing Symposium (IGARSS), pp. 2091–2094 (2013)
31. Riccardi, A., Fernndez-Navarro, F., Carloni, S.: Cost-sensitive adaboost algorithm for ordinal regression based on extreme learning machine. IEEE Transactions on Cybernetics **44**(10), 1898–1909 (2014)
32. Sun, Y., Kamel, M.S., Wong, A.K., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. Pattern Recognition **40**(12), 3358–3378 (2007)
33. Tan, M.: Cost-sensitive learning of classification knowledge and its applications in robotics. Machine Learning **13**(1), 7–33 (1993)
34. Ting, K.M.: An instance-weighting method to induce cost-sensitive trees. Knowledge and Data Engineering, IEEE Transactions on **14**(3), 659–665 (2002)
35. Viola, P., Jones, M.J.: Robust real-time face detection. International journal of computer vision **57**(2), 137–154 (2004)
36. Wan, S., Ye, T., Li, M., Zhang, H., Li, X.: Efficient spherical parametrization using progressive optimization. In: Computational Visual Media, pp. 170–177. Springer (2012)
37. Wan, S., Yin, Z., Zhang, K., Zhang, H., Li, X.: A topology-preserving optimization algorithm for polycube mapping. Computers & Graphics **35**(3), 639–649 (2011)
38. Wang, H., Wang, J.: An effective image representation method using kernel classification. In: Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on, pp. 853–858 (2014)
39. Wang, J., Zhao, P., Hoi, S.: Cost-sensitive online classification. In: Proceedings - IEEE International Conference on Data Mining, ICDM, pp. 1140–1145 (2012)

40. Wang, S., Zhou, Y., Tan, J., Xu, J., Yang, J., Liu, Y.: Computational modeling of magnetic nanoparticle targeting to stent surface under high gradient field. Computational mechanics **53**(3), 403–412 (2014)
41. Xu, L., Zhan, Z., Xu, S., Ye, K.: An evasion and counter-evasion study in malicious websites detection. In: Communications and Network Security (CNS), 2014 IEEE Conference on, pp. 265–273. IEEE (2014)
42. Xu, S., Lu, W., Xu, L., Zhan, Z.: Adaptive epidemic dynamics in networks: Thresholds and control. ACM Transactions on Autonomous and Adaptive Systems (TAAS) **8**(4), 19 (2014)
43. Xu, S., Lu, W., Zhan, Z.: A stochastic model of multivirus dynamics. Dependable and Secure Computing, IEEE Transactions on **9**(1), 30–45 (2012)
44. X.Wang, C.Kambhamettu: Gender classification of depth images based on shape and texture analysis. In: GlobalSIP, pp. 1077–1080 (2013)
45. X.Wang, V.Ly, G.Guo, C.Kambhamettu: A new approach for 2d-3d heterogeneous face recognition. In: ISM, pp. 301–304 (2013)
46. X.Wang, V.Ly, G.Lu, C.Kambhamettu: Can we minimize the influence due to gender and race in age estimation? In: 2013 12th International Conference on Machine Learning and Applications (ICMLA), vol. 2, pp. 309–314 (2013)
47. Yang, J., Yan, R., Hauptmann, A.: Cross-domain video concept detection using adaptive svms. In: Proceedings of the ACM International Multimedia Conference and Exhibition, pp. 188–197 (2007)
48. Zadrozny, B., Langford, J., Abe, N.: Cost-sensitive learning by cost-proportionate example weighting. In: Proceedings - IEEE International Conference on Data Mining, ICDM, pp. 435–442 (2003)
49. Zaina, L., Ameida, T., Torres, G.: Can the online social networks be used as a learning tool? a case study in twitter. Communications in Computer and Information Science **446 CCIS**, 114–123 (2014)
50. Zeng, J., Beth, P.: Data pipeline in mapreduce. In: 2013 IEEE 9th International Conference on eScience, eScience '13, pp. 164–171. IEEE (2013)
51. Zeng, J., Ruan, G., Crowell, A., Prakash, A., Plale, B.: Cloud computing data capsules for non-consumptiveuse of texts. In: Proceedings of the 5th ACM Workshop on Scientific Cloud Computing, ScienceCloud '14, pp. 9–16. ACM (2014)
52. Zhan, Z., Xu, M., Xu, S.: A characterization of cybersecurity posture from network telescope data. In: Proceedings of The 6th International Conference on Trustworthy Systems, InTrust, vol. 14
53. Zhan, Z., Xu, M., Xu, S.: Characterizing honeypot-captured cyber attacks: Statistical framework and case study. Information Forensics and Security, IEEE Transactions on **8**(11), 1775–1789 (2013)
54. Zhao, X., Miao, S., Du, L., Liao, R.: Robust 2-d/3-d registration of ct volumes with contrast-enhanced x-ray sequences in electrophysiology based on a weighted similarity measure and sequential subspace optimization. In: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pp. 934–938 (2013)
55. Zhao, X., Wang, Y., Jozsef, G.: Robust shape-constrained active contour for whole heart segmentation in 3-d ct images for radiotherapy planning. In: Image Processing (ICIP), 2014 IEEE International Conference on, pp. 1–5 (2014)
56. Zhao, X., Wong, E.K., Wang, Y., Lymberis, S., Wen, B., Formenti, S., Chang, J.: A support vector machine (svm) for predicting preferred treatment position in radiotherapy of patients with breast cancer. Medical Physics **37**(10), 5341–5350 (2010)
57. Zhou, Y., Hu, W., Peng, B., Liu, Y.: Biomarker binding on an antibody-functionalized biosensor surface: The influence of surface properties, electric field, and coating density. The Journal of Physical Chemistry C **118**(26), 14,586–14,594 (2014)
58. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. Knowledge and Data Engineering, IEEE Transactions on **18**(1), 63–77 (2006)
59. Zhu, H., Dwyer, M.B., Goddard, S.: Predictable runtime monitoring. In: Real-Time Systems, 2009. ECRTS'09. 21st Euromicro Conference on, pp. 173–183. IEEE (2009)
60. Zhu, H., Goddard, S., Dwyer, M.: Selecting server parameters for predictable runtime monitoring. In: Real-Time and Embedded Technology and Applications Symposium (RTAS), 2010 16th IEEE, pp. 227–236. IEEE (2010). DOI 10.1109/RTAS.2010.18

61. Zhu, H., Goddard, S., Dwyer, M.B.: Response time analysis of hierarchical scheduling: The synchronized deferrable servers approach. In: Real-Time Systems Symposium (RTSS), 2011 IEEE 32nd, pp. 239–248. IEEE (2011)


Average classifcation accuracy
1
0.9
0.8
0.7
0.6
OCSCA
ABC
CSB
STM
SW


Average misclassification cost
0.4
0.3
0.2
0.1
0
OCSCA
ABC
CSB
STM
SWW